The Journal of Open Source Software

Nashpy: A Python library for the computation of Nash equilibria

Vincent Knight[1] **and James Campbell**[1]

1 Cardiff University, School of Mathematics, UK

DOI: 10.21105/joss.00904

Software

- Review
- Repository
- Archive

Submitted: 31 May 2018
Published: 10 October 2018

Summary

Game theory is the study of strategic interactions where the outcomes of choice depend on the choices of all participants. A key solution concept in the field is that of Nash Equilibrium (Nash & others, 1950). This solution concept corresponds to a coordinate at which no participant has any incentive to change their choice.

As an example, consider the game of Rock Paper Scissors, which can be represented mathematically using the following matrix:

$$A = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}$$

The rows and columns correspond to the actions available: Rock, Paper and Scissors. A value of 1 indicates that that specific row beats the corresponding column and similarly a value of -1 indicates a loss and a 0 indicates a tie. For example, A_{21} shows that Paper (the second action) beats Rock (the first action). Using `Nashpy`, the equilibrium behaviour can be computed:

```
>>> import nashpy as nash
>>> import numpy as np
>>> A = np.array([[0, -1, 1], [1, 0, -1], [-1, 1, 0]])
>>> game = nash.Game(A)
>>> for eq in game.support_enumeration():
...     print(eq)
(array([ 0.33..., 0.33..., 0.33...]), array([ 0.33..., 0.33..., 0.33...]))
```

As expected: both players should play each action randomly (each with probability 1/3).

Computing these equilibria for large games, where individuals have many strategic options available to them, requires the use of software implementations of known algorithms. A number of algorithms exist to compute these Nash equilibria, for example the Lemke-Howson algorithm (Lemke & Howson, 1964).

Statement of need

Access to these algorithms is non trival, an example is the modelling of healthcare decisions (Vincent Knight, Komenda, & Griffiths, 2017) where a bespoke theoretic result was used to design a specific algorithm for the computation of equilibria. Accessible software would make that research more straightforward as no new algorithm would need to be implemented.

The most mature piece of software available for the computation of equilibria is **Gambit** (McKelvey, McLennan, & Turocy, 2006). Gambit has a Python wrapper to its core C functionality however is not currently portable. For example Windows is not supported. There does exist a web interface with a Gambit back end: Game theory explorer however this is not practical for reproducible research.

`Nashpy` is a Python library with all dependencies being part of the standard scientific Python stack—NumPy and SciPy (Jones, Oliphant, Peterson, & others, 2001)—thus it is portable. For example, Windows support is regularly tested through a Windows continuous integration service (Appveyor).

`Nashpy` currently implements 3 algorithms for the computation of equilibria (currently only for 2-player games) and is extensively documented, including theoretic reference material on the algorithms: nashpy.readthedocs.io. Furthermore, the software is automatically tested using a combination of doc (this paper is also tested), unit, integration and property based tests with 100% coverage.

Potential limitations of `Nashpy` are due to the complexity of the algorithms themselves. For example, support enumeration enumerates all potential pairs of strategies. For $n \times n$ square matrices it has $\mathcal{O}\left(2^{n^2}\right)$ complexity. All implementations provided in `Nashpy` ensure these effects are reduced: NumPy (Jones et al., 2001) provides C based implementations for vectorized performance. Furthermore, all algorithms are generators, which ensures that not all equilibria must be found before one is returned. For example, below, an 11-by-11 game is considered and timings are shown for relative comparison. Using the more efficient Lemke-Howson algorithm (Lemke & Howson, 1964), an equilibrium is found approximately 3000 times faster.

```
>>> from pprint import pprint
>>> A = np.eye(11)
>>> game = nash.Game(A, A[::-1])
>>> pprint(next(game.support_enumeration()))  # 2.26 s ± 118 ms per loop
(array([ 0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.]),
 array([ 0.,  0.,  0.,  0.,  0.,  1.,  0.,  0.,  0.,  0.,  0.]))
>>> pprint(next(game.lemke_howson_enumeration()))  # 734 µs ± 5.27 µs per loop
(array([ 0.5,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0.5]),
 array([ 0.5,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0. ,  0.5]))
```

`Nashpy` is designed to be used by researchers but also students in courses in the fields of mathematics, computer science and/or economics. It is currently being used in a final-year course at Cardiff University. Due to the fact that the code is written entirely in Python and is open source, this makes it a positive teaching tool as students can read and understand the implementation of the algorithms. `Nashpy` has been archived to Zenodo (Vince Knight & Baldevia, 2018).

Acknowledgements

We acknowledge code contributions from Ria Baldevia as well as many helpful discussions with Nikoleta Glynatsi.

We would also like to the thank the reviewers and editor for their comments and suggestions which helped improve this manuscript.

References

Jones, E., Oliphant, T., Peterson, P., & others. (2001). SciPy: Open source scientific tools for Python. Retrieved from http://www.scipy.org/

Knight, V., & Baldevia, R. (2018, January). Drvinceknight/nashpy: V0.0.13. doi:10.5281/zenodo.1163694

Knight, V., Komenda, I., & Griffiths, J. (2017). Measuring the price of anarchy in critical care unit interactions. *Journal of the Operational Research Society*, *68*(6), 630–642. doi:10.1057/s41274-016-0100-8

Lemke, C. E., & Howson, J. T., Jr. (1964). Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, *12*(2), 413–423. doi:10.1137/0112033

McKelvey, R. D., McLennan, A. M., & Turocy, T. L. (2006). Gambit: Software tools for game theory.

Nash, J. F., & others. (1950). Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, *36*(1), 48–49. doi:10.1073/pnas.36.1.48